# ----------------------------------------------------------

# Information Retrieval Evaluation Exercise

# Python version (equivalent to the provided R script)

# ----------------------------------------------------------

import os

import pandas as pd

import numpy as np

from scipy.stats import pearsonr, ttest\_rel

#--Run this cell only if you’re using Google Colab; otherwise, skip it.---

from google.colab import drive

drive.mount('/content/drive')

# ---------- CONFIGURATION ----------

QRELS\_FILE = "/content/drive/MyDrive/TRECDATA/qrels.trec8.adhoc" # path to qrels file

RUNS\_FOLDER = "/content/drive/MyDrive/TRECDATA/" # folder containing ~15 system runs

TOP\_K = 10

# ---------- STEP 1: LOAD QRELS ----------

def load\_qrels(filepath):

"""Load TREC qrels file."""

qrels = pd.read\_csv(filepath, sep=r"\s+", header=None, names=["topic", "iter", "docid", "relevance"])

return qrels

qrels = load\_qrels(QRELS\_FILE)

# ---------- STEP 2: LOAD SYSTEM RUN ----------

def load\_run(set20):

"""Load one system run in standard TREC 6-column format."""

run = pd.read\_csv(set20, sep=r"\s+", header=None,

names=["topic", "Q0", "docid", "rank", "score", "system"])

return run

# ---------- STEP 3: COMPUTE PRECISION@K ----------

def precision\_at\_k(run, qrels, k=10):

"""Compute Precision@k per topic."""

topics = run["topic"].unique()

results = []

for t in topics:

docs = run[run["topic"] == t].sort\_values("rank").head(k)["docid"].tolist()

relevant\_docs = qrels[(qrels["topic"] == t) & (qrels["relevance"] > 0)]["docid"].tolist()

precision = len(set(docs) & set(relevant\_docs)) / k

results.append({"topic": t, "Pk": precision})

return pd.DataFrame(results)

# ---------- STEP 4: LOOP THROUGH FILES INSIDE set20 ----------

all\_scores = pd.DataFrame()

SET\_FOLDER = os.path.join(RUNS\_FOLDER, "set20")

files = os.listdir(SET\_FOLDER)

print("Files in set20:", files)

for f in files:

run\_path = os.path.join(SET\_FOLDER, f)

# skip directories and non-run files

if os.path.isdir(run\_path):

continue

if not f.startswith("input."):

print(f"⚠️ Skipping non-run file: {f}")

continue

print(f"📂 Evaluating: {f}")

run = load\_run(run\_path)

p10\_df = precision\_at\_k(run, qrels, TOP\_K)

p10\_df["system"] = f

all\_scores = pd.concat([all\_scores, p10\_df], ignore\_index=True)

# ---------- STEP 5: COMPUTE AVERAGE P@10 ----------

avg\_scores = all\_scores.groupby("system")["Pk"].mean().reset\_index(name="avg\_P10")

avg\_scores = avg\_scores.sort\_values("avg\_P10", ascending=False)

print("=== Average Precision@10 for set20 ===")

print(avg\_scores.to\_string(index=False))

